B 7

```
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
```

**F

FILEID**CNFSHOW   F 2

```
  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  HH      HH    OOOOOO    WW      WW
  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  HH      HH    OOOOOO    WW      WW
CC          NN      NN  FF          SS          HH      HH  OO      OO  WW      WW
CC          NN      NN  FF          SS          HH      HH  OO      OO  WW      WW
CC          NNNN    NN  FF          SS          HH      HH  OO      OO  WW      WW
CC          NNNN    NN  FF          SS          HH      HH  OO      OO  WW      WW
CC          NN  NN  NN  FFFFFFFF      SSSSSS    HHHHHHHHHH  OO      OO  WW      WW
CC          NN  NN  NN  FFFFFFFF      SSSSSS    HHHHHHHHHH  OO      OO  WW      WW
CC          NN    NNNN  FF                  SS  HH      HH  OO      OO  WW  WW  WW
CC          NN    NNNN  FF                  SS  HH      HH  OO      OO  WW  WW  WW
CC          NN      NN  FF                  SS  HH      HH  OO      OO  WWWW  WWWW      ....
CC          NN      NN  FF                  SS  HH      HH  OO      OO  WWWW  WWWW      ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS    HH      HH    OOOOOO    WW      WW      ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS    HH      HH    OOOOOO    WW      WW      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE  'DECnet Ethernet Configurator Module'
0 MODULE CNFSHOW          (
0                         LANGUAGE (BLISS32),
0                         IDENT = 'V04-000'
0                         ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     DECnet Configurator Module (NICONFIG)
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the routines to return information on a
1 !       SHOW request generated by an NCP> SHOW MODULE CONFIGUTOR command.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  13-Oct-1982
1 !
1 ! MODIFIED BY:
1 !
1 !--
```

```
%SBTTL  'Definitions'

!
! INCLUDE FILES:
!

LIBRARY 'SYS$LIBRARY:STARLET';  ! VMS common definitions

LIBRARY 'SHRLIB$:NMALIBRY';     ! NICE code definitions

REQUIRE 'LIB$:CNFDEF.R32';

REQUIRE 'SRC$:CNFPREFIX.REQ';


!
! BUILTIN functions
!

BUILTIN
    SUBM;                       ! To support quadword subtraction

LITERAL
    NICE_BUFLEN = 128;

!
! TABLE OF CONTENTS:
!

FORWARD ROUTINE

        PROCESS_SHOW,           ! Cover routine for common error handling of SHOW processing
        SHOW_CIRCUIT,           ! Format circuit info
        SHOW_SYSTEM;            ! Format info for a system ID message.

!
! EXTERNAL REFERENCES:
!

EXTERNAL ROUTINE

    ! Module CNFMAIN

        CNF$EXIT,               ! Clean up and exit
        CNF$TRACE,              ! Log messages to log file
        CNF$FREE_VM,            ! Free virtual memory
        CNF$GET_ZVM,            ! Get zeroed virtual memory

    ! Module CNFREQUES

        CNF$LOCATE_CIR_BLK,     ! Locate circuit block from circuit name

    ! Module CNFSEND

        CNF$BUFR_NICE_MSG,      ! Buffer NICE response messages
        CNF$BUFR_ERR_MSG;       ! Buffer NICE error responses
```

```
EXTERNAL

        CNF$GQ_CIRSURLST : VECTOR [2];  ! List of circuits under surveillance

OWN
    NICE_DONE_DSC :                     ! NICE 'DONE' message
        $BBLOCK [DSC$C_S_BLN] INITIAL
        (
            1,
            UPLIT (
                BYTE (%X'80')
                )
        ),

    NICE_MORE_DSC :                     ! NICE 'MORE' message
        $BBLOCK [DSC$C_S_BLN] INITIAL
        (
            4,
            UPLIT (
                BYTE (%X'02'),
                WORD (%X'FFFF'),
                BYTE (%X'00')
                )
        );
```

```
1 %SBTTL  'CNF$PROCESS_SHOW   Search the data base and format a response message'
1 GLOBAL ROUTINE CNF$PROCESS_SHOW (IRB, KNOWN, CIRCUITNAM_DSC, INFTYP) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !  Shell routine to supply a common entrance and error exit to the
1 !  routine which builds the SHOW message.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       irb             Interrupt request block, contains context for returning
1 !                       responses to connectee.
1 !
1 !       known           Was SHOW KNOWN CIRCUITS requested?
1 !
1 !       circuitnam_dsc  Descriptor of circuit name if SHOW was requested for
1 !                       a specific circuit.
1 !
1 !       inftyp          Code determining which information type was requested
1 !                       for the SHOW, either CHARACTERISTICS, SUMMARY or STATUS.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !  Always success, errors are buffered for return to connectee.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
2     BEGIN
2     LOCAL
2         STATUS;
2
2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
2         $DESCRIPTOR ('CNF$PROCESS_SHOW'));
2
2     !
2     !   Send MORE message
2     !
2     EXECUTE (CNF$BUFR_NICE_MSG (.IRB, NICE_MORE_DSC, 0));
2
2     !
2     !   Request that the SHOW information be gathered, formatted and buffered.
2     !
2     STATUS = PROCESS_SHOW (.IRB, .KNOWN, .CIRCUITNAM_DSC, .INFTYP);
2     IF NOT .STATUS
2     THEN
2         CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_MPR, 0, .STATUS);
2
```

```
;   188          0372  2        !
;   189          0373  2        !   Send DONE message
;   190          0374  2        !
;   191          0375  2        EXECUTE (CNF$BUFR_NICE_MSG (.IRB, NICE_DONE_DSC, 0));
;   192          0376  2
;   193          0377  2        RETURN TRUE;
;   194          0378  1        END;                              ! Routine CNF$PROCESS_SHOW


                                                                          .TITLE  CNFSHOW DECnet Ethernet Configurator Module
                                                                          .IDENT  \V04-000\

                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                    80  00000 P.AAA:     .BYTE   -128                                         ;
                                                        00001            .BLKB   3
                                                    02  00004 P.AAB:     .BYTE   2                                            ;
                                                  FFFF  00005            .WORD   -1                                           ;
                                                    00  00007            .BYTE   0                                            ;
                                    45  43  41  52  54  00008 P.AAD:     .ASCII  \TRACE\                                      ;
                                                        0000D            .BLKB   3
                                              00000005  00010 P.AAC:     .LONG   5                                            ;
                                             00000000'  00014            .ADDRESS P.AAD                                       ;
4F  48  53  5F  53  53  45  43  4F  52  50  24  46  4E  43  00018 P.AAF:     .ASCII  \CNF$PROCESS_SHOW\                           ;
                                                    57  00027                                                                 ;
                                              00000010  00028 P.AAE:     .LONG   16                                           ;
                                             00000000'  0002C            .ADDRESS P.AAF                                       ;

                                                                          .PSECT  $OWN$,NOEXE,2

                                              00000001  00000 NICE_DONE_DSC:
                                                                          .LONG   1
                                             00000000'  00004            .ADDRESS P.AAA                                       ;
                                              00000004  00008 NICE_MORE_DSC:
                                                                          .LONG   4                                            ;
                                             00000000'  0000C            .ADDRESS P.AAB                                       ;

                                                                          .EXTRN  CNF$EXIT, CNF$TRACE
                                                                          .EXTRN  CNF$FREE_VM, CNF$GET_ZVM
                                                                          .EXTRN  CNF$LOCATE_CIR_BLK
                                                                          .EXTRN  CNF$BUFR_NICE_MSG
                                                                          .EXTRN  CNF$BUFR_ERR_MSG
                                                                          .EXTRN  CNF$GQ_CIRSURLST

                                                                          .PSECT  $CODE$,NOWRT,2

                                                  0000  00000            .ENTRY  CNF$PROCESS_SHOW, Save nothing               ; 0316
                                     0000'   CF   9F    00002            PUSHAB  P.AAE                                        ; 0357
                                     0000'   CF   9F    00006            PUSHAB  P.AAC                                        ; 0356
                                             01   DD    0000A            PUSHL   #1                                           ;
                        0000G   CF           03   FB    0000C            CALLS   #3, CNF$TRACE                                ;
                                             7E   D4    00011            CLRL    -(SP)                                        ; 0362
                                     0000'   CF   9F    00013            PUSHAB  NICE_MORE_DSC                                ;
                                        04   AC   DD    00017            PUSHL   IRB                                          ;
                        0000G   CF           03   FB    0001A            CALLS   #3, CNF$BUFR_NICE_MSG                        ;
                                33           50   E9    0001F            BLBC    STATUS, 2$                                   ;
                                7E      0C   AC   7D    00022            MOVQ    CIRCUITNAM_DSC, -(SP)                        ; 0367
```

```
                          7E          04    AC  7D 00026             MOVQ     IRB, -(SP)                                       ;
                   0000V  CF                04  FB 0002A             CALLS    #4, PROCESS_SHOW                                 ;
                          0F                50  E8 0002F             BLBS     STATUS, 1$                                       ; 0368
                                            50  DD 00032             PUSHL    STATUS                                           ; 0370
                                            7E  D4 00034             CLRL     -(SP)                                            ;
                          7E                05  CE 00036             MNEGL    #5, -(SP)                                        ;
                                      04    AC  DD 00039             PUSHL    IRB                                              ;
                   0000G  CF                04  FB 0003C             CALLS    #4, CNF$BUFR_ERR_MSG                             ;
                                            7E  D4 00041 1$:         CLRL     -(SP)                                            ; 0375
                                   0000'    CF  9F 00043             PUSHAB   NICE_DONE_DSC                                    ;
                                      04    AC  DD 00047             PUSHL    IRB                                              ;
                   0000G  CF                03  FB 0004A             CALLS    #3, CNF$BUFR_NICE_MSG                            ;
                          03                50  E9 0004F             BLBC     STATUS, 2$                                       ;
                          50                01  D0 00052             MOVL     #1, R0                                           ; 0377
                                                04 00055 2$:         RET                                                       ; 0378

; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL  'process_show   Search the data base and format a response message'
1 ROUTINE PROCESS_SHOW (IRB, KNOWN, CIRCUITNAM_DSC, INFTYP) =
1
1 !++
1 !  Locate requested circuit or dispatch for all known circuits to
1 !  the routine which will format and buffer the SHOW response.
1 !
1 !        irb              Interrupt request block, contains context for returning
1 !                         responses to connectee.
1 !
1 !        known            Was SHOW KNOWN CIRCUITS requested?
1 !
1 !        circuitnam_dsc   Descriptor of circuit name if SHOW was requested for
1 !                         a specific circuit.
1 !
1 !        inftyp           Code determining which information type was requested
1 !                         for the SHOW, either CHARACTERISTICS, SUMMARY or STATUS.
1 !
1 !  Always return success, any errors will be buffered for return to
1 !  connectee.
1 !--
1
2     BEGIN
2     MAP
2         CIRCUITNAM_DSC : REF BBLOCK;
2
2     LOCAL
2         CIR : REF BBLOCK;
2
2
2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
2         $DESCRIPTOR ('process_show'));
2
2     IF .KNOWN
2     THEN
2         !
2         !    Format the data for all circuits
2         !
3         BEGIN
3         CIR = .CNF$GQ_CIRSURLST;                  ! List of circuits under surveillance
3         WHILE .CIR NEQ CNF$GQ_CIRSURLST DO        ! For the entire list of circuits
4             BEGIN
4             EXECUTE (SHOW_CIRCUIT (.IRB, .CIR, .INFTYP));
4             CIR = .CIR [CIR$L_LINK];              ! Get next circuit in list
3             END;                                  ! While traversing list of circuits
2         END
2     ELSE
3         BEGIN
3         !
3         !    Locate the requested circuit and format the data for it.
3         !
3         IF CNF$LOCATE_CIR_BLK (.CIRCUITNAM_DSC, CIR)
3         THEN
4             EXECUTE (SHOW_CIRCUIT (.IRB, .CIR, .INFTYP))
3         ELSE
4             BEGIN                                 ! Oops, that circuit is not in the data base
4             CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_IDE, NMA$C_ENT_CIR, 0, .CIRCUITNAM_DSC);
```

```
: 253           0436  4                 RETURN TRUE;
: 254           0437  3                 END;
: 255           0438  3
: 256           0439  2             END;
: 257           0440  2
: 258           0441  2         RETURN TRUE;
: 259           0442  1         END;                            ! Routine process_show


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                           45  43  41  52  54  00030 P.AAH:     .ASCII  \TRACE\                                         ;
                                               00035            .BLKB   3
                                     00000005  00038 P.AAG:     .LONG   5                                               ;
                                     00000000' 0003C            .ADDRESS P.AAH                                          ;
    77  6F  68  73  5F  73  73  65  63  6F  72  70  00040 P.AAJ:     .ASCII  \process_show\                                  ;
                                     0000000C  0004C P.AAI:     .LONG   12                                              ;
                                     00000000' 00050            .ADDRESS P.AAJ                                          ;


                                                                .PSECT  $CODE$,NOWRT,2

                                          0000 00000 PROCESS_SHOW:
                                                                .WORD   Save nothing                                    ; 0380
                          5E           04  C2 00002            SUBL2   #4, SP                                          ;
                                 0000'  CF  9F 00005            PUSHAB  P.AAI                                           ; 0410
                                 0000'  CF  9F 00009            PUSHAB  P.AAG                                           ; 0409
                                        01  DD 0000D            PUSHL   #1                                              ;
                    0000G  CF           03  FB 0000F            CALLS   #3, CNF$TRACE                                   ;
                           24      08   AC  E9 00014            BLBC    KNOWN, 2$                                       ; 0412
                           6E   0000G   CF  D0 00018            MOVL    CNF$GQ_CIRSURLST, CIR                           ; 0418
                           50   0000G   CF  9E 0001D 1$:        MOVAB   CNF$GQ_CIRSURLST, R0                            ; 0419
                           50           6E  D1 00022            CMPL    CIR, R0                                         ;
                                        45  13 00025            BEQL    4$                                              ;
                                   10   AC  DD 00027            PUSHL   INFTYP                                          ; 0421
                                   04   AE  DD 0002A            PUSHL   CIR                                             ;
                                   04   AC  DD 0002D            PUSHL   IRB                                             ;
                    0000V  CF           03  FB 00030            CALLS   #3, SHOW_CIRCUIT                                ;
                           37           50  E9 00035            BLBC    STATUS, 5$                                      ;
                                        9E  DD 00038            PUSHL   @CIR                                            ; 0422
                                        E1  11 0003A            BRB     1$                                              ; 0419
                                        5E  DD 0003C 2$:        PUSHL   SP                                              ; 0430
                                   0C   AC  DD 0003E            PUSHL   CIRCUITNAM_DSC                                  ;
                    0000G  CF           02  FB 00041            CALLS   #2, CNF$LOCATE_CIR_BLK                          ;
                           12           50  E9 00046            BLBC    R0, 3$                                          ;
                                   10   AC  DD 00049            PUSHL   INFTYP                                          ; 0432
                                   04   AE  DD 0004C            PUSHL   CIR                                             ;
                                   04   AC  DD 0004F            PUSHL   IRB                                             ;
                    0000V  CF           03  FB 00052            CALLS   #3, SHOW_CIRCUIT                                ;
                           12           50  E8 00057            BLBS    STATUS, 4$                                      ;
                                            04 0005A            RET                                                     ;
                                   0C   AC  DD 0005B 3$:        PUSHL   CIRCUITNAM_DSC                                  ; 0435
                           7E           03  7D 0005E            MOVQ    #3, -(SP)                                       ;
                           7E           09  CE 00061            MNEGL   #9, -(SP)                                       ;
                                   04   AC  DD 00064            PUSHL   IRB                                             ;
```

```
                            0000G  CF             05  FB 00067            CALLS    #5, CNF$BUFR_ERR_MSG                           ;
                                   50             01  D0 0006C 4$:        MOVL     #1, R0                                          ; 0441
                                                      04 0006F 5$:        RET                                                      ; 0442

; Routine Size:  112 bytes,    Routine Base:  $CODE$ + 0056
```

```
%SBTTL  'show circuit   Format all systems for circuit'
ROUTINE SHOW_CIRCUIT (IRB, CIR, INFTYP) =

!++
!  Build the NICE for the SHOW response message and buffer it for
!  transmission to the connectee.
!
!       irb             Interrupt request block, contains context for returning
!                       responses to connectee.
!
!       cir             Address of Circuit control block of circuit SHOW
!                       was requested for.
!
!       inftyp          Code determining which information type was requested
!                       for the SHOW, either CHARACTERISTICS, SUMMARY or STATUS.
!
!  Always return success, any errors will be buffered for return to
!  connectee.
!--

    BEGIN
    MAP
        CIR : REF BBLOCK;

    LOCAL
        CURRENT_TIMBUF : BBLOCK [8],            ! Buffer to obtain the current system time
        DELTA_TIMBUF : BBLOCK [8],              ! Buffer to calculate the time difference between the curren
                                                !  the time surveillance began on the circuit.
        NICE : REF BBLOCK,                      ! Pointer into the buffer where the NICE message is being bu
        NICE_BUFDSC : BBLOCK [DSC$C_S_BLN],     ! Descriptor of NICE message buffer
        NICE_TMPDSC : BBLOCK [DSC$C_S_BLN],     ! Descriptor of NICE Template buffer
        SID : REF BBLOCK,                       ! Pointer to a system ID message
        TIMBUF : VECTOR [7, WORD];              ! Buffer for converting binary time format to ASCII for NICE

    BIND
        CONF = UPLIT (%ASCIC 'CONFIGURATOR') : VECTOR [,BYTE];  ! Module name to place into NICE return


    CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
        $DESCRIPTOR ('show_circuit'));

    !
    !   Zero the descriptor which will locate the buffer where the NICE response
    !   will be built, allocate the buffer, and initialize buffer pointer.
    !
    CH$FILL (0, DSC$C_S_BLN, NICE_TMPDSC);
    EXECUTE (CNF$GET_ZVM (%REF (NICE_BUFLEN), NICE_TMPDSC [DSC$A_POINTER]));
    NICE = .NICE_TMPDSC [DSC$A_POINTER];

    !
    !   Place Error status
    !
    !       1 byte      return code
    !       2 bytes     error detail
    !       1 byte      length of error message
    !
    (.NICE) <0, 8> = %X'01';                    ! Return code SUCCESS
```

```
2       (.NICE) <8, 16> = %X'FFFF';                     ! Error detail, SUCCESS
2       (.NICE) <24, 8> = %X'00';                       ! Error text length
2
2       !
2       !   Copy over the module entity, CONFIGURATOR
2       !
2       !       1 byte      Length of CONFIGURATOR string
2       !       12 bytes    CONFIGURATOR string
2       !
2       (.NICE) <32, 8> = .CONF [0];                    ! Length of CONFIGURATOR string
2       NICE = .NICE + 5;                               ! Set pointer to beginning of circuit name
2       CH$MOVE (.CONF [0], CONF [1], .NICE);
2       NICE_TMPDSC [DSC$W_LENGTH] = 5 + .CONF [0];
2       NICE = .NICE + .CONF [0];                       ! Point to free space in buffer after
2                                                       !  the circuit name which was just copied in
2
2       !
2       !   Copy over Circuit name entity
2       !
2       !       2 bytes     Circuit entity ID
2       !       1 byte      Parameter type = ASCII
2       !       1 byte      Length of circuit name
2       !       n bytes     Circuit name
2       !
2       (.NICE) <0, 16> = NMA$C_PCCN_CIR;
2       (.NICE) <16, 8> = NMA$C_PTY_AI;                 ! Type = ASCII
2       (.NICE) <24, 8> = .CIR [CIR$W_CIRNAMLEN];       ! Length of Circuit name
2       NICE = .NICE + 4;                               ! Set pointer to beginning of circuit name
2       CH$MOVE (.CIR [CIR$W_CIRNAMLEN], CIR [CIR$T_CIRNAM], .NICE);
2       NICE = .NICE + .CIR [CIR$W_CIRNAMLEN];          ! Point to free space in buffer after
2                                                       !  the circuit name which was just copied in
2       NICE_TMPDSC [DSC$W_LENGTH] = .NICE_TMPDSC [DSC$W_LENGTH] + 4 + .CIR [CIR$W_CIRNAMLEN];
2
2       !
2       !   Place in Surveillance parameter
2       !   as a coded value
2       !
2       !       2 bytes     Surviellance parameter ID
2       !       1 byte      Surveillance type = coded byte
2       !       1 byte      Surveillance value
2       !
2       (.NICE) <0, 16> = NMA$C_PCCN_SUR;
3           BEGIN
3           BIND
3               TYPE = .NICE + 2 : BBLOCK;
3           TYPE [NMA$V_PTY_COD] = TRUE;                ! Surveillance is returned as a coded value
3           TYPE [NMA$V_PTY_CLE] = 1;                   ! The coded value is 1 byte in length
3           END;
2       (.NICE) <24, 8> = .CIR [CIR$B_SURVEIL];
2       NICE = .NICE + 4;                               ! Set pointer to end of buffer where Elapsed Time will be pl
2
2       !
2       !   Place in Elapsed Time parameter
2       !   as a coded multiple
2       !
2       !       2 bytes     Elasped Time parameter ID
2       !       1 byte      Elasped time type = coded multiple of 3 fields
```

```
2                  !       1 byte      hours type = unsigned decimal word
2                  !       2 bytes     hours value
2                  !       1 byte      minutes type = unsigned decimal byte
2                  !       1 byte      minutes value
2                  !       1 byte      seconds type = unsigned decimal byte
2                  !       1 byte      seconds value
2                  !
2           (.NICE) <0, 16> = NMA$C_PCCN_ELT;                     ! Set parameter ID
3               BEGIN
3               BIND
3                   CODMUL_TYP = .NICE + 2 : BBLOCK,
3                   HR_TYP = .NICE + 3 : BBLOCK,
3                   MIN_TYP = .NICE + 6 : BBLOCK,
3                   SEC_TYP = .NICE + 8 : BBLOCK;
3
3               CODMUL_TYP [NMA$V_PTY_COD] = TRUE;                ! Elapsed Time is returned as a coded
3               CODMUL_TYP [NMA$V_PTY_MUL] = TRUE;                !  multiple.
3               CODMUL_TYP [NMA$V_PTY_CLE] = 3;                   ! There are three fields in the coded multiple
3
3               !
3               !    Get the current system time, subtract
3               !    Time of Surveillance start from Current time
3               !    to get negative Delta time
3               !
3               EXECUTE ($GETTIM (TIMADR = CURRENT_TIMBUF));
3               SUBM (2, CIR [CIR$Q_ELAPSDTIM], CURRENT_TIMBUF, DELTA_TIMBUF);
3               EXECUTE ($NUMTIM (TIMBUF = TIMBUF, TIMADR = DELTA_TIMBUF));
3
3               HR_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;             ! Unsigned decimal
3               HR_TYP [NMA$V_PTY_NLE] = 2;                        !  word.
3               (.NICE) <32, 16> = .TIMBUF [3];                    ! Hours
3
3               MIN_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;            ! Unsigned decimal
3               MIN_TYP [NMA$V_PTY_NLE] = 1;                       !  byte.
3               (.NICE) <56, 8> = .TIMBUF [4];                     ! Minutes
3
3               SEC_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;            ! Unsigned decimal
3               SEC_TYP [NMA$V_PTY_NLE] = 1;                       !  byte.
3               (.NICE) <72, 8> = .TIMBUF [5];                     ! Seconds
2               END;
2
2           NICE_TMPDSC [DSC$W_LENGTH] = 14 + .NICE_TMPDSC [DSC$W_LENGTH];
2
2           SID = .CIR [CIR$L_SIDFLINK];                          ! Point to first System ID
2
2           IF (.SID EQL CIR [CIR$L_SIDFLINK]) OR                 ! There are no ID's collected for this circuit
3               ((.INFTYP NEQ NMA$C_OPINF_STA) AND                ! or Summary requested
3                   (.INFTYP NEQ NMA$C_OPINF_CHA))
2           THEN
2               !
2               !   Print only circuit info, not system ID's, since either
2               !   there are no ID's collected, or a SHOW SUMMARY was requested.
2               !
3               BEGIN
3               CNF$BUFR_NICE_MSG (.IRB, NICE_TMPDSC, NICE_BUFLEN);
3               RETURN TRUE;
3               END
```

```
3
2       ELSE
2           !
2           !   Traverse the list of system ID's and format a NICE response
2           !   for each one.  Each one will be appended to a repeat of the
2           !   circuit info already gathered.
2           !
3           BEGIN
3           WHILE .SID NEQ CIR [CIR$L_SIDFLINK] DO          ! For all the System ID's
4               BEGIN
4               !
4               !   Zero the descriptor and allocate a clean buffer for the NICE
4               !   response message.  Then copy the message already built for the
4               !   circuit info as the start of the message to which the system ID
4               !   info will be appended.
4               !
4               CH$FILL (0, DSC$C_S_BLN, NICE_BUFDSC);
4               EXECUTE (CNF$GET_ZVM (%REF (NICE_BUFLEN), NICE_BUFDSC [DSC$A_POINTER]));
4               CH$MOVE (.NICE_TMPDSC [DSC$W_LENGTH], .NICE_TMPDSC [DSC$A_POINTER],
4                   .NICE_BUFDSC [DSC$A_POINTER]);
4               NICE_BUFDSC [DSC$W_LENGTH] = .NICE_TMPDSC [DSC$W_LENGTH];
4
4               !
4               !   Append the system ID info to the NICE response, and
4               !   buffer the message for later transmission.
4               !   Then follow list pointer to next system ID.
4               !
4               SHOW_SYSTEM (.SID, NICE_BUFDSC);
4               CNF$BUFR_NICE_MSG (.IRB, NICE_BUFDSC, NICE_BUFLEN);
4               SID = .SID [SID$L_LINK];
3               END;                            ! While processing all system ID's for the circuit
3           !
3           !   Return the buffer which we used to build the circuit info part
3           !   of the response.
3           !
3           EXECUTE (CNF$FREE_VM (%REF (NICE_BUFLEN), NICE_TMPDSC [DSC$A_POINTER]));
2           END;                                ! There are system ID's for this circuit
2
2       RETURN TRUE;
1       END;                                    ! Routine show_circuit


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

00  00  52  4F  54  41  52  55  47  49  46  4E  4F  43  0C  00054 P.AAK:  .ASCII  <12>\CONFIGURATOR\<0><0><0>
                                                        00  00063
                                        45  43  41  52  54  00064 P.AAM:  .ASCII  \TRACE\
                                                            00069         .BLKB   3
                                                  00000005  0006C P.AAL:  .LONG   5
                                                  00000000' 00070         .ADDRESS P.AAM
        74  69  75  63  72  69  63  5F  77  6F  68  73  00074 P.AAO:  .ASCII  \show_circuit\
                                                  0000000C  00080 P.AAN:  .LONG   12
                                                  00000000' 00084         .ADDRESS P.AAO

                                                            CONF=                   P.AAK
                                                                  .EXTRN  SYS$GETTIM, SYS$NUMTIM
```

```
                                                               .PSECT   $CODE$,NOWRT,2

                                            00FC 00000 SHOW_CIRCUIT:
                                                               .WORD    Save R2,R3,R4,R5,R6,R7                        ; 0444
                                5E        34  C2 00002         SUBL2    #52, SP
                                   0000'  CF  9F 00005         PUSHAB   P.AAN                                         ; 0482
                                   0000'  CF  9F 00009         PUSHAB   P.AAL                                         ; 0481
                                          01  DD 0000D         PUSHL    #1
                         0000G  CF        03  FB 0000F         CALLS    #3, CNF$TRACE
      08          00            6E        00  2C 00014         MOVC5    #0, (SP), #0, #8, NICE_TMPDSC                 ; 0488
                                      14  AE     00019
                                      18  AE  9F 0001B         PUSHAB   NICE_TMPDSC+4                                 ; 0489
                            04  AE    80  8F  9A 0001E         MOVZBL   #128, 4(SP)
                                      04  AE  9F 00023         PUSHAB   4(SP)
                         0000G  CF        02  FB 00026         CALLS    #2, CNF$GET_ZVM
                                79        50  E9 0002B         BLBC     STATUS, 1$
                                56    18  AE  D0 0002E         MOVL     NICE_TMPDSC+4, NICE                           ; 0490
                                86        01  90 00032         MOVB     #1, (NICE)+                                   ; 0499
                                86        01  AE 00035         MNEGW    #1, (NICE)+                                   ; 0500
                                          86  94 00038         CLRB     (NICE)+                                       ; 0501
                                57 0000'  CF  9A 0003A         MOVZBL   CONF, R7                                      ; 0509
                                86        57  90 0003F         MOVB     R7, (NICE)+
                  66     0000'  CF        57  28 00042         MOVC3    R7, CONF+1, (NICE)                            ; 0511
             14   AE            57        05  A1 00048         ADDW3    #5, R7, NICE_TMPDSC                           ; 0512
                                56        57  C0 0004D         ADDL2    R7, NICE                                      ; 0513
                                86    64  8F  9B 00050         MOVZBW   #100, (NICE)+                                 ; 0524
                                86    40  8F  90 00054         MOVB     #64, (NICE)+                                  ; 0525
                                57    08  AC  D0 00058         MOVL     CIR, R7                                       ; 0526
                                86    16  A7  90 0005C         MOVB     22(R7), (NICE)+
                  66        18  A7    16  A7  28 00060         MOVC3    22(R7), 24(R7), (NICE)                        ; 0528
                                50    16  A7  32 00066         CVTWL    22(R7), R0                                    ; 0529
                                56        50  C0 0006A         ADDL2    R0, NICE
                                50    14  AE  3C 0006D         MOVZWL   NICE_TMPDSC, R0                               ; 0531
                                51    16  A7  32 00071         CVTWL    22(R7), R1
                                50        51  C0 00075         ADDL2    R1, R0
             14   AE            50        04  A1 00078         ADDW3    #4, R0, NICE_TMPDSC
                                86    6E  8F  9B 0007D         MOVZBW   #110, (NICE)+                                 ; 0541
                                66    80  8F  88 00081         BISB2    #128, (NICE)                                  ; 0545
      B6          06            00        01  F0 00085         INSV     #1, #0, #6, (NICE)+                           ; 0546
                                86    0A  A7  90 0008A         MOVB     10(R7), (NICE)+                               ; 0548
                                66    6F  8F  9B 0008E         MOVZBW   #111, (NICE)                                  ; 0564
                            02  A6    C0  8F  88 00092         BISB2    #192, 2(NICE)                                 ; 0573
02    A6          06            00        03  F0 00097         INSV     #3, #0, #6, 2(NICE)                           ; 0574
                                      2C  AE  9F 0009D         PUSHAB   CURRENT_TIMBUF                                ; 0581
                    000000000G  00        01  FB 000A0         CALLS    #1, SYS$GETTIM
                                1E        50  E9 000A7 1$:     BLBC     STATUS, 2$
             24   AE        2C  AE    30  A7  C3 000AA         SUBL3    48(R7), CURRENT_TIMBUF, DELTA_TIMBUF          ; 0582
                            28  AE    30  AE  D0 000B1         MOVL     CURRENT_TIMBUF, DELTA_TIMBUF
                            28  AE    34  A7  D9 000B6         SBWC     52(R7), DELTA_TIMBUF
                                      24  AE  9F 000BB         PUSHAB   DELTA_TIMBUF                                  ; 0583
                                      08  AE  9F 000BE         PUSHAB   TIMBUF
                    000000000G  00        02  FB 000C1         CALLS    #2, SYS$NUMTIM
                                7B        50  E9 000C8 2$:     BLBC     STATUS, 5$
                            03  A6        30  8A 000CB         BICB2    #48, 3(NICE)                                  ; 0585
03    A6          04            00        02  F0 000CF         INSV     #2, #0, #4, 3(NICE)                           ; 0586
                            04  A6    0A  AE  B0 000D5         MOVW     TIMBUF+6, 4(NICE)                             ; 0587
```

```
                                   06   A6         30  8A 000DA         BICB2   #48, 6(NICE)                                  ; 0590
     06   A6             04             00         01  F0 000DE         INSV    #1, #0, #4, 6(NICE)                           ; 0590
                                   07   A6   0C    AE  90 000E4         MOVB    TIMBUF+8, 7(NICE)                             ; 0591
                                   08   A6         30  8A 000E9         BICB2   #48, 8(NICE)                                  ; 0593
     08   A6             04             00         01  F0 000ED         INSV    #1, #0, #4, 8(NICE)                           ; 0594
                                   09   A6   0E    AE  90 000F3         MOVB    TIMBUF+10, 9(NICE)                            ; 0595
                                   14   AE         0E  A0 000F8         ADDW2   #14, NICE_TMPDSC                              ; 0598
                                        56   40    A7  D0 000FC         MOVL    64(R7), SID                                   ; 0600
                                        50   40    A7  9E 00100         MOVAB   64(R7), R0                                    ; 0602
                                        50         56  D1 00104         CMPL    SID, R0
                                                   0C  13 00107         BEQL    3$
                                        01   0C    AC  D1 00109         CMPL    INFTYP, #1                                    ; 0603
                                                   17  13 0010D         BEQL    4$
                                        02   0C    AC  D1 0010F         CMPL    INFTYP, #2                                    ; 0604
                                                   11  13 00113         BEQL    4$
                                        7E   80    8F  9A 00115 3$:     MOVZBL  #128, -(SP)                                   ; 0611
                                             18    AE  9F 00119         PUSHAB  NICE_TMPDSC
                                             04    AC  DD 0011C         PUSHL   IRB
                                0000G   CF         03  FB 0011F         CALLS   #3, CNF$BUFR_NICE_MSG
                                                   60  11 00124         BRB     7$                                            ; 0612
                                        50   40    A7  9E 00126 4$:     MOVAB   64(R7), R0                                    ; 0622
                                        50         56  D1 0012A         CMPL    SID, R0
                                                   44  13 0012D         BEQL    6$
          08             00             6E         00  2C 0012F         MOVC5   #0, (SP), #0, #8, NICE_BUFDSC                 ; 0630
                                             1C    AE     00134
                                             20    AE  9F 00136         PUSHAB  NICE_BUFDSC+4                                 ; 0631
                                   04   AE   80    8F  9A 00139         MOVZBL  #128, 4(SP)
                                             04    AE  9F 0013E         PUSHAB  4(SP)
                                0000G   CF         02  FB 00141         CALLS   #2, CNF$GET_ZVM
                                        40         50  E9 00146 5$:     BLBC    STATUS, 8$
                    20   BE        18   BE   14    AE  28 00149         MOVC3   NICE_TMPDSC, @NICE_TMPDSC+4, @NICE_BUFDSC+4   ; 0633
                                   1C   AE   14    AE  B0 00150         MOVW    NICE_TMPDSC, NICE_BUFDSC                      ; 0634
                                             1C    AE  9F 00155         PUSHAB  NICE_BUFDSC                                   ; 0641
                                                   56  DD 00158         PUSHL   SID
                                0000V   CF         02  FB 0015A         CALLS   #2, SHOW_SYSTEM
                                        7E   80    8F  9A 0015F         MOVZBL  #128, -(SP)                                   ; 0642
                                             20    AE  9F 00163         PUSHAB  NICE_BUFDSC
                                             04    AC  DD 00166         PUSHL   IRB
                                0000G   CF         03  FB 00169         CALLS   #3, CNF$BUFR_NICE_MSG
                                        56         66  D0 0016E         MOVL    (SID), SID                                    ; 0643
                                                   B3  11 00171         BRB     4$                                            ; 0622
                                             18    AE  9F 00173 6$:     PUSHAB  NICE_TMPDSC+4                                 ; 0649
                                   04   AE   80    8F  9A 00176         MOVZBL  #128, 4(SP)
                                             04    AE  9F 0017B         PUSHAB  4(SP)
                                0000G   CF         02  FB 0017E         CALLS   #2, CNF$FREE_VM
                                        03         50  E9 00183         BLBC    STATUS, 8$
                                        50         01  D0 00186 7$:     MOVL    #1, R0                                        ; 0652
                                                       04 00189 8$:     RET                                                   ; 0653

; Routine Size:  394 bytes,    Routine Base:  $CODE$ + 00C6
```

```
1 %SBTTL  'show_system   Format System ID info'
1 ROUTINE SHOW_SYSTEM (SID, NICEBUF) =
1
1 !++
1 !   Format the information in the system ID message stored in
1 !   SID and build a NICE message which will be appended to the
1 !   NICE message for the circuit which is in NICEBUF.
1 !
1 !       sid       Pointer to buffer containing a system ID message
1 !
1 !       nicebuf   Descriptor of buffer containing circuit NICE message
1 !
1 !   Always return success.  There is no error checking.
1 !
1 !--
2     BEGIN
2     MAP
2         NICEBUF : REF BBLOCK,
2         SID : REF BBLOCK;
2     LOCAL
2         NICE : REF BBLOCK,
2         TIMBUF : VECTOR [7, WORD];
2
2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
2         $DESCRIPTOR ('show_system'));
2
2     NICE = .NICEBUF [DSC$A_POINTER] + .NICEBUF [DSC$W_LENGTH];
2
2     !
2     !   Place in Physical Address parameter
2     !   as a Hex Image 6
2     !
2     !       2 bytes     Physical Address parameter ID
2     !       1 byte      Physical Address type = Hex Image (HI-6)
2     !       6 bytes     Physical Address value
2     !
2     (.NICE) <0, 16> = NMA$C_PCCN_PHA;
3         BEGIN
3         BIND
3             TYPE = .NICE + 2 : BBLOCK;
3         TYPE [NMA$V_PTY_NTY] = NMA$C_NTY_H;     ! returned as a Hex
3         TYPE [NMA$V_PTY_NLE] = NMA$C_NLE_IMAGE; !    image.
2         END;
2     (.NICE) <24,8> = SID$C_ADRLEN;
2     CH$MOVE (SID$C_ADRLEN, SID [SID$T_CURADR], (.NICE + 4) );
2     NICE = .NICE + 4 + SID$C_ADRLEN;             ! Set pointer to end of buffer where next parameter will be
```

```
2       !
2       !   Place in Last Report parameter
2       !   as a coded multiple
2       !
2       !       2 bytes     Last Report parameter ID
2       !       1 byte      Last Report type = coded multiple of 5 fields
2       !       1 byte      Day type = unsigned decimal byte
2       !       1 byte      Day value
2       !       1 byte      Month type = Coded byte
2       !       1 byte      Month coded value
2       !       1 byte      hour type = unsigned decimal byte
2       !       1 byte      hour value
2       !       1 byte      minutes type = unsigned decimal byte
2       !       1 byte      minutes value
2       !       1 byte      seconds type = unsigned decimal byte
2       !       1 byte      seconds value
2       !
2       (.NICE) <0, 16> = NMA$C_PCCN_LRP;
3           BEGIN
3           BIND
3               CODMUL_TYP = .NICE + 2 : BBLOCK,
3               DAY_TYP = .NICE + 3 : BBLOCK,
3               MON_TYP = .NICE + 5 : BBLOCK,
3               HR_TYP = .NICE + 7 : BBLOCK,
3               MIN_TYP = .NICE + 9 : BBLOCK,
3               SEC_TYP = .NICE + 11 : BBLOCK;
3
3           CODMUL_TYP [NMA$V_PTY_COD] = TRUE;                  ! Last Report is returned as a coded
3           CODMUL_TYP [NMA$V_PTY_MUL] = TRUE;                  !  multiple.
3           CODMUL_TYP [NMA$V_PTY_CLE] = 5;                     ! There are five fields in the coded multiple
3
3           EXECUTE ($NUMTIM (TIMBUF = TIMBUF, TIMADR = SID [SID$Q_LSTREPORT]) );
3
3           DAY_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU; ! Unsigned decimal
3           DAY_TYP [NMA$V_PTY_NLE] = 1;            !  byte.
3           (.NICE) <32, 8> = .TIMBUF [2];          ! Day
3
3           MON_TYP [NMA$V_PTY_COD] = TRUE;         ! Month is returned as a coded value
3           MON_TYP [NMA$V_PTY_CLE] = 1;            !   contained in 1 byte.
3           (.NICE) <48, 8> = .TIMBUF [1];          ! Month
3
3           HR_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;  ! Unsigned decimal
3           HR_TYP [NMA$V_PTY_NLE] = 1;             !  byte.
3           (.NICE) <64, 8> = .TIMBUF [3];          ! Hour
3
3           MIN_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU; ! Unsigned decimal
3           MIN_TYP [NMA$V_PTY_NLE] = 1;            !  byte.
3           (.NICE) <80, 8> = .TIMBUF [4];          ! Minute
3
3           SEC_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU; ! Unsigned decimal
3           SEC_TYP [NMA$V_PTY_NLE] = 1;            !  byte.
3           (.NICE) <96, 8> = .TIMBUF [5];          ! Second
2           END;
2       NICE = .NICE + 13;
```

```
2     !
2     !   Place in Maintenance Version parameter
2     !   as a coded multiple
2     !
2     !       2 bytes     Maintenance Version parameter ID
2     !       1 byte      Maintenance Version type = coded multiple of 3 fields
2     !       1 byte      Version Number type = unsigned decimal byte
2     !       1 byte      Version Number value
2     !       1 byte      ECO number type = unsigned decimal byte
2     !       1 byte      ECO number value
2     !       1 byte      User ECO number type = unsigned decimal byte
2     !       1 byte      User ECO value
2     !
2     (.NICE) <0, 16> = NMA$C_PCCN_MVR;
3         BEGIN
3         BIND
3             CODMUL_TYP = .NICE + 2 : BBLOCK,
3             VN_TYP = .NICE + 3 : BBLOCK,
3             EN_TYP = .NICE + 5 : BBLOCK,
3             UEN_TYP = .NICE + 7 : BBLOCK;
3
3         CODMUL_TYP [NMA$V_PTY_COD] = TRUE;              ! Maintenance Version is returned as a coded
3         CODMUL_TYP [NMA$V_PTY_MUL] = TRUE;              !  multiple.
3         CODMUL_TYP [NMA$V_PTY_CLE] = 3;                 ! There are three fields in the coded multiple
3
3         VN_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;          ! Unsigned decimal
3         VN_TYP [NMA$V_PTY_NLE] = 1;                     !  byte.
3         (.NICE) <32, 8> = .SID [SID$B_MOPVER];          ! MOP version
3
3         EN_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;          ! Unsigned decimal
3         EN_TYP [NMA$V_PTY_NLE] = 1;                     !  byte.
3         (.NICE) <48, 8> = .SID [SID$B_MOPECO];          ! MOP ECO
3
3         UEN_TYP [NMA$V_PTY_NTY] = NMA$C_NTY_DU;         ! Unsigned decimal
3         UEN_TYP [NMA$V_PTY_NLE] = 1;                    !  byte.
3         (.NICE) <64, 8> = .SID [SID$B_MOPUSRECO];
2         END;
2
2     NICE = .NICE + 9;
2
```

```
2       !
2       !       Place in Functions parameter
2       !       as a coded multiple
2       !
2       !           2 bytes     Functions parameter ID
2       !           1 byte      Functions type = coded multiple of up to 16 fields
2       !           n bytes     Function type = Coded byte
2       !
2       !           up to 16 functions permitted
2       !
2
2       IF .SID [SID$B_NUMFUNC] NEQ 0
2       THEN
3           BEGIN
3           BIND
3               CODMUL_TYP = .NICE + 2 : BBLOCK,
3               FUNCTIONS = SID [SID$W_FUNCTIONS] : BITVECTOR [16];
3
3           (.NICE) <0, 16> = NMA$C_PCCN_FCT;                   ! Place in Function paramter ID
3           CODMUL_TYP [NMA$V_PTY_COD] = TRUE;                  ! Report is returned as a coded
3           CODMUL_TYP [NMA$V_PTY_MUL] = TRUE;                  !   multiple.
3
3           CODMUL_TYP [NMA$V_PTY_CLE] = .SID [SID$B_NUMFUNC];      ! 16 fields are permitted in the coded multi
3           NICE = .NICE + 3;
3
3           INCR INDEX FROM 0 TO SID$C_MAXFUNC - 1 DO
4               BEGIN
4               BIND
4                   FUN_TYP = .NICE : BBLOCK;
4
4               IF .FUNCTIONS [.INDEX]
4               THEN
5                   BEGIN
5                   FUN_TYP [NMA$V_PTY_COD] = TRUE;             ! Functions are returned as a coded value
5                   FUN_TYP [NMA$V_PTY_CLE] = 1;                !   contained in 1 byte.
5                   (.NICE) <8, 8> = .INDEX;                    ! Place Function value in NICE message
5                   NICE = .NICE + 2;                           ! Advance to end of NICE buffer
4                   END;
3               END;
2           END;
```

```
;   658          0835  2        !
;   659          0836  2        !   Place in Hardware Address parameter
;   660          0837  2        !   as a Hex Image 6
;   661          0838  2        !
;   662          0839  2        !       2 bytes     Hardware Address parameter ID
;   663          0840  2        !       1 byte      Hardware Address type = Hex Image (HI-6)
;   664          0841  2        !       1 byte      Image length = 6
;   665          0842  2        !       6 bytes     Hardware Address value
;   666          0843  2        !
;   667          0844  2        (.NICE) <0, 16> = NMA$C_PCCN_HWA;
;   668          0845  3            BEGIN
;   669          0846  3            BIND
;   670          0847  3                TYPE = .NICE + 2 : BBLOCK;
;   671          0848  3            TYPE [NMA$V_PTY_NTY] = NMA$C_NTY_H;                ! returned as a Hex
;   672          0849  3            TYPE [NMA$V_PTY_NLE] = NMA$C_NLE_IMAGE;            !   image
;   673          0850  2            END;
;   674          0851  2        (.NICE) <24,8> = SID$C_ADRLEN;                          !   of length 6
;   675          0852  2        CH$MOVE (SID$C_ADRLEN, SID [SID$T_HRDWADR], (.NICE + 4) );
;   676          0853  2        NICE = .NICE + 4 + SID$C_ADRLEN;                        ! Set pointer to end of buffer where next parameter
;   677          0854  2
;   678          0855  2
;   679          0856  2        !
;   680          0857  2        !   Place in Device Type parameter
;   681          0858  2        !   as a coded value
;   682          0859  2        !
;   683          0860  2        !       2 bytes     Device Type parameter ID
;   684          0861  2        !       1 byte      Device Type type = coded byte
;   685          0862  2        !       1 byte      Device Type code
;   686          0863  2        !
;   687          0864  2        (.NICE) <0, 16> = NMA$C_PCCN_DTY;
;   688          0865  3            BEGIN
;   689          0866  3            BIND
;   690          0867  3                TYPE = .NICE + 2 : BBLOCK;
;   691          0868  3            TYPE [NMA$V_PTY_COD] = TRUE;            ! Device Type is returned as a coded value
;   692          0869  3            TYPE [NMA$V_PTY_CLE] = 1;               ! The coded value is 1 byte in length
;   693          0870  2            END;
;   694          0871  2        (.NICE) <24, 8> = .SID [SID$B_DEVICE];
;   695          0872  2        NICE = .NICE + 4;                           ! Set pointer to end of buffer where Elapsed Time will be pl
;   696          0873  2
;   697          0874  2        NICEBUF [DSC$W_LENGTH] = .NICE - .NICEBUF [DSC$A_POINTER];
;   698          0875  2
;   699          0876  2        RETURN TRUE;
;   700          0877  1        END;                             ! Routine show_system


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                               45  43  41  52  54  00088 P.AAQ:   .ASCII  \TRACE\                                  ;
                                                   0008D          .BLKB   3                                        ;
                                        00000005   00090 P.AAP:   .LONG   5                                        ;
                                        00000000'  00094          .ADDRESS P.AAQ                                   ;
         6D  65  74  73  79  73  5F  77  6F  68  73  00098 P.AAS:  .ASCII  \show_system\                            ;
                                                   000A3          .BLKB   1                                        ;
                                        0000000B   000A4 P.AAR:   .LONG   11                                       ;
                                        00000000'  000A8          .ADDRESS P.AAS                                   ;
```

```
                                                                .PSECT   $CODE$,NOWRT,2

                                           01FC   00000 SHOW_SYSTEM:
                                                                .WORD    Save R2,R3,R4,R5,R6,R7,R8     ; 0655
                               5E          10  C2 00002         SUBL2    #16, SP
                                    0000'  CF  9F 00005         PUSHAB   P.AAR                         ; 0678
                                    0000'  CF  9F 00009         PUSHAB   P.AAP                         ; 0677
                                           01  DD 0000D         PUSHL    #1
                        0000G  CF          03  FB 0000F         CALLS    #3, CNF$TRACE
                               57      04  AC  7D 00014         MOVQ     SID, R7                       ; 0698
                               56          68  3C 00018         MOVZWL   (R8), NICE                    ; 0680
                               56      04  A8  C0 0001B         ADDL2    4(R8), NICE
                               66      78  8F  9B 0001F         MOVZBW   #120, (NICE)                  ; 0690
02  A6         02              04          02  F0 00023         INSV     #2, #4, #2, 2(NICE)           ; 0694
                           02  A6          0F  8A 00029         BICB2    #15, 2(NICE)                  ; 0695
                           03  A6          06  90 0002D         MOVB     #6, 3(NICE)                   ; 0697
          04   A6          10  A7          06  28 00031         MOVC3    #6, 16(R7), 4(NICE)           ; 0698
                               56          0A  C0 00037         ADDL2    #10, NICE                     ; 0699
                               66      82  8F  9B 0003A         MOVZBW   #130, (NICE)                  ; 0717
                           02  A6      C0  8F  88 0003E         BISB2    #192, 2(NICE)                 ; 0728
02  A6         06              00          05  F0 00043         INSV     #5, #0, #6, 2(NICE)           ; 0729
                                       16  A7  9F 00049         PUSHAB   22(R7)                        ; 0731
                                       04  AE  9F 0004C         PUSHAB   TIMBUF
                    00000000G  00          02  FB 0004F         CALLS    #2, SYS$NUMTIM
                               01          50  E8 00056         BLBS     STATUS, 1$
                                               04 00059         RET
                           03  A6          30  8A 0005A 1$:     BICB2    #48, 3(NICE)                  ; 0733
03  A6         04              00          01  F0 0005E         INSV     #1, #0, #4, 3(NICE)           ; 0734
                           04  A6      04  AE  90 00064         MOVB     TIMBUF+4, 4(NICE)             ; 0735
                           05  A6      80  8F  88 00069         BISB2    #128, 5(NICE)                 ; 0737
05  A6         06              00          01  F0 0006E         INSV     #1, #0, #6, 5(NICE)           ; 0738
                           06  A6      02  AE  90 00074         MOVB     TIMBUF+2, 6(NICE)             ; 0739
                           07  A6          30  8A 00079         BICB2    #48, 7(NICE)                  ; 0741
07  A6         04              00          01  F0 0007D         INSV     #1, #0, #4, 7(NICE)           ; 0742
                           08  A6      06  AE  90 00083         MOVB     TIMBUF+6, 8(NICE)             ; 0743
                           09  A6          30  8A 00088         BICB2    #48, 9(NICE)                  ; 0745
09  A6         04              00          01  F0 0008C         INSV     #1, #0, #4, 9(NICE)           ; 0746
                           0A  A6      08  AE  90 00092         MOVB     TIMBUF+8, 10(NICE)            ; 0747
                           0B  A6          30  8A 00097         BICB2    #48, 11(NICE)                 ; 0749
0B  A6         04              00          01  F0 0009B         INSV     #1, #0, #4, 11(NICE)          ; 0750
                           0C  A6      0A  AE  90 000A1         MOVB     TIMBUF+10, 12(NICE)           ; 0751
                               56          0D  C0 000A6         ADDL2    #13, NICE                     ; 0753
                               86    4E21  8F  B0 000A9         MOVW     #20001, (NICE)+               ; 0768
                               66      C0  8F  88 000AE         BISB2    #192, (NICE)                  ; 0777
    86         06              00          03  F0 000B2         INSV     #3, #0, #6, (NICE)+           ; 0778
                               66          30  8A 000B7         BICB2    #48, (NICE)                   ; 0780
    86         04              00          01  F0 000BA         INSV     #1, #0, #4, (NICE)+           ; 0781
                               86      1E  A7  90 000BF         MOVB     30(R7), (NICE)+               ; 0782
                               66          30  8A 000C3         BICB2    #48, (NICE)                   ; 0784
    86         04              00          01  F0 000C6         INSV     #1, #0, #4, (NICE)+           ; 0785
                               86      1F  A7  90 000CB         MOVB     31(R7), (NICE)+               ; 0786
                               66          30  8A 000CF         BICB2    #48, (NICE)                   ; 0788
    86         04              00          01  F0 000D2         INSV     #1, #0, #4, (NICE)+           ; 0789
                               86      20  A7  90 000D7         MOVB     32(R7), (NICE)+               ; 0790
                                       21  A7  95 000DB         TSTB     33(R7)                        ; 0806
```

```
                                                                 B  4
CNFSHOW          DECnet Ethernet Configurator Module        16-Sep-1984 02:05:37    VAX-11 Bliss-32 V4.0-742              Page  22
V04-000          show_system    Format System ID info       14-Sep-1984 12:49:54    [NICNF.SRC]CNFSHOW.B32;1                  (10)

                                                     26  13 000DE        BEQL    4$                                ;
                                     86      4E22    8F  B0 000E0        MOVW    #20002, (NICE)+                   ; 0813
                                     66        C0    8F  88 000E5        BISB2   #192, (NICE)                      ; 0815
           86              06        00        21    A7  F0 000E9        INSV    33(R7), #0, #6, (NICE)+           ; 0817
                                                     50  D4 000EF        CLRL    INDEX                             ; 0820
                           0C      22  A7            50  E1 000F1  2$:   BBC     INDEX, 34(R7), 3$                 ; 0825
                                     66        80    8F  88 000F6        BISB2   #128, (NICE)                      ; 0828
           86              06        00              01  F0 000FA        INSV    #1, #0, #6, (NICE)+               ; 0829
                                     86              50  90 000FF        MOVB    INDEX, (NICE)+                    ; 0830
                           EB        50              0F  F3 00102  3$:   AOBLEQ  #15, INDEX, 2$                    ; 0820
                                     66      4E27    8F  B0 00106  4$:   MOVW    #20007, (NICE)                    ; 0844
     02    A6              02        04              02  F0 0010B        INSV    #2, #4, #2, 2(NICE)               ; 0848
                                   02  A6            0F  8A 00111        BICB2   #15, 2(NICE)                      ; 0849
                                   03  A6            06  90 00115        MOVB    #6, 3(NICE)                       ; 0851
                    04     A6      0A  A7            06  28 00119        MOVC3   #6, 10(R7), 4(NICE)               ; 0852
                                     56              0A  C0 0011F        ADDL2   #10, NICE                         ; 0853
                                     86      4E84    8F  B0 00122        MOVW    #20100, (NICE)+                   ; 0864
                                     66        80    8F  88 00127        BISB2   #128, (NICE)                      ; 0868
           86              06        00              01  F0 0012B        INSV    #1, #0, #6, (NICE)+               ; 0869
                                     86        24    A7  90 00130        MOVB    36(R7), (NICE)+                   ; 0871
                           68        56        04    A8  A3 00134        SUBW3   4(R8), NICE, (R8)                 ; 0874
                                     50              01  D0 00139        MOVL    #1, R0                            ; 0876
                                                         04 0013C        RET                                       ; 0877

; Routine Size:  317 bytes,    Routine Base:  $CODE$ + 0250
;  702          0878  1 END                             ! End of module CNFSHOW
;  703          0879  0 ELUDOM



;                                         PSECT SUMMARY

        Name                        Bytes                        Attributes

   $PLIT$                               172  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
   $OWN$                                 16  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
   $CODE$                               909  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                              Library Statistics

                                             -------- Symbols --------        Pages        Processing
        File                                 Total   Loaded   Percent        Mapped        Time

   _$255$DUA28:[SYSLIB]STARLET.L32;1          9776        8         0          581          00:01.0
   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1          887       23         2           47          00:00.7



;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CNFSHOW/OBJ=OBJ$:CNFSHOW MSRC$:CNFSHOW/UPDATE=(ENH$:CNFSHOW)

; Size:           909 code + 188 data bytes
; Run Time:          00:23.1
; Elapsed Time:      00:39.7
; Lines/CPU Min:      2282
; Lexemes/CPU-Min: 19848
; Memory Used:  182 pages
; Compilation Complete
```

0280 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNFSEND
LIS

NPADEF
MDL

CNFSTORE
LIS

NMLSHR
MAP

NMAHEAD
B32

NPAMAC
MAR

CNFSHOW
LIS

CNFWQDEF
LIS

NMATAIL
B32

NML

NMLDDL
B32

NML
MAP

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS